12-4-2013

Security Of Internet Protocol Cameras – A Case Example

William Campbell
Edith Cowan University, wocampbe@our.ecu.edu.au

DOI: 10.4225/75/57b3c06efb868
11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

SECURITY OF INTERNET PROTOCOL CAMERAS – A CASE EXAMPLE

William Campbell
Edith Cowan University
wocampbe@our.ecu.edu.au

Abstract

The interaction of consumer devices and the internet, especially in relation to security, has always been tenuous. Where it is in the best interests of companies to produce products that are cheap and accessible, these traits often go against that of security. This investigation undertakes an analysis of one such device – the DCS-930L internet protocol camera from D-Link. This camera is analysed for vulnerabilities, with an emphasis on those relating to authentication mechanisms. Several vulnerabilities are identified, and potential attacks based on these are discussed. Solutions or mitigations to these vulnerabilities are presented.

Keywords
IP Camera, authentication, dlink, DCS-930L, embedded device

INTRODUCTION

Internet protocol cameras are small electronic devices, present in home, commercial and business variants. They typically consist of a video camera attached to a small web server, allowing access to the device via internet protocols. This access can be used to view the cameras vision, as well as to update and adjust its settings. Unfortunately, there have been recent suggestions that IP cameras contain less than ideal security (Heffner, 2013a; Sood & Gajbhiye, 2011). Given their purpose as security devices, this can be understandably upsetting for consumers as their privacy maybe impacted.

This research studied one such device: the DCS-930L from D-Link (D-Link Australia, 2013). This camera was selected due to its features, availability and low cost. It is considered to be representative of the consumer-grade IP camera market in general. This brand of camera also possess a history of dangerous vulnerabilities, including authentication bypasses and command injection (Paleari, 2013; Rocha, M., et al., 2013).

In terms of features, the camera contains the ability to wirelessly access a network, as well as operate as its own access point. Additionally, it can collaborate with online software controlled by D-Link, to give access to the device from remote locations. The device can also be programmed to allow remote access over the internet. As of November 2013, there are over 40,000 DCS-930L cameras directly accessible by internet users (Shodan, 2013).

The investigation was broken into several phases. Firstly, information regarding how the IP camera operates, including the contents of its embedded system, will be researched. This process will focus on the authentication mechanisms in particular. From here, possible attacks against these authentication processes will be analysed. Following this, post-authenticated attacks will be investigated. That is, attacks that require an authenticated user to perform.

METHODOLOGY

A means to monitor and manipulate communication between the IP camera and the interacting computer was required. Wireshark was used to monitor the network stream, during both Ethernet and wireless communications (Wireshark Foundation, 2013). Burpsuite was employed to capture traffic moving toward the camera, and manipulate it as necessary (PortSwigger, 2013).

The contents of the current firmware embedded in the camera was considered vital pieces of information. As such, they were unpacked using a program called binwalk (Heffner, 2013b). This

provided information regarding the resources held on the device, including those employed by the web server. Additionally, certain open source software used in the device employ certain licences for example for video compression. These licences state that a copy of the source code must be made available to interested parties if the programs are used in commercial systems. A copy of this information was obtained, but found to be outdated.

Web based forms, and other methods through which to communicate with the device, were checked for sanitisation. Put another way, the web server was tested to see if it properly escapes and sanitises the information it receives. If this is not the case, the camera may be open to manipulation. HTML and hex encoding schemes were used, as well as special characters (such as null and terminating bytes).

Authentication and Pre Authentication Attacks

The DCS-930L camera in question appears to utilise two different authentication schemes: basic access and digest access (NWG, 1999). Basic access authentication consists of encoding the password and username using the base 64 encoding scheme. This result is then bundled with an authorisation header, and attached to any requests for resources. A web server will then reverse the encoding and check the credentials against its own list.

Digest access is slightly more complex, in this case involving md5 hashing. Several different pieces of information are combined and hashed in a specified manner. This information includes a password and username, as well as a realm, uri request and created nonce. The outcome of this process, along with the specified nonce, is sent in response to an authorisation request. If the same hash value is achieved by the server, it is assumed that the same username and password was used in each calculation (and thus that the client is authorised).

Each of these authentication schemes carries with it some weaknesses. Firstly, the basic authentication scheme contains no real protection in terms of confidentiality. Any member of the network that has access to the traffic stream will be able to reverse the base64 encoding. This will reveal the plaintext values for the username and password. Base64 character encoding was designed as a means to transfer information, rather than protect it.

The digest authentication method is considerably safer in terms of confidentiality, however, still has some issues. Firstly, the current implementation in the camera appears to allow replay attacks. A user who has managed to capture a legitimate authorisation request, can simply resend that request themselves, to 'authenticate', and gain access to a resource.

That said, this result is somewhat limited given the design of digest authentication. One of the parameters of this access scheme is that the URL being requested by a party is itself hashed within the request. This means that any replay attacks will only be effective against a single resource or request. In relation to the camera, there are instances where the resource being requested is the video feed, and thus a replay attack against this resource will be particularly effective. It is theorised that an attacker can continually replay the authenticated request for constant access to this feed.

Alternatively (or additionally), an eavesdropping attacker may wish to conduct an offline attack against a legitimate digest authorisation request. As it stands, four of the five pieces of information required for digest authentication are known to an eavesdropping attacker (ie the username is locked as ‘admin’, the realm, the uri requested and the nonce). From here, it would be fairly simple to create a brute force script to iterate through possible values for the password, and check these against the expected hash.

The possibility of an online brute force attack is also available against the DCS-930L. As the username for the system is defaulted to 'admin', only the password is unknown to an attacker. From here, an attacker is able to generate a possible password, encode it depending on the scheme required (ie. basic or digest access) and send a request to the cameras server. There does not appear to be a

mechanism to prevent this kind of attack built into the camera, such as a lockout period after a certain number of failed attempts. Additionally, there does not appear to be any logging or reporting for this kind of behaviour.

This threat of a brute force attack is significantly increased when considering an additional bug within the older firmware versions. Specifically, until the most recent patch (Firmware v1.08b04), passwords for the DSC-930L were limited to only eight characters (D-Link, 2013). This significantly decreases the number of combinations for a password, and thus increases the efficacy of brute forcing the device.

Another attack against the DCS-930L authentication may be possible through UDP broadcasts. The camera in question comes with a wizard to help during set-up and installation. Analysis of the communication between the camera and this wizard appears to show that a UDP broadcast is sent by the wizard, which the client camera responds to (also in the form of a UDP broadcast). As these are broadcast packets (as opposed to direct) any member of the subnet may pick them up.

There is evidence to suggest that these transmissions contain the password for the camera (Doyle, 2012), albeit in what appears to be a custom encryption scheme.

This encryption scheme may be susceptible to a limited range, known plain text attack. Specifically, the camera name appears to be one of the pieces of information sent during this exchange. An attacker may choose to manipulate this variable, and monitor the differences it makes on the encrypted traffic. In addition, it appears that this encrypted communication occurs without the passing of any variables (such as session keys). In turn, this would suggest that the algorithm for encryption is hard coded into a program, stored both on the device and the wizard application. If an attacker were to reverse engineer this software, they may have alternate means for determining how the encryption scheme works. From here, they may be able to manipulate the requests for information to the device, and thus compromise its security.

Reverse engineering of the firmware also reveals a piece of information intriguing to an attacker: a public and private key pair. Although the server appears to support SSL traffic, this was only seen during communication with the official D-Link website (when attempting to access the device remotely through their interface). Applying the private key with Wireshark, some of the communication between the camera and the website was revealed. This communication appeared to access resources that were already publically accessible (see the next paragraph for more information on this). No further information was revealed within this communication stream, but it does suggest that the camera can be directly communicated with by D-Link.

In addition to the weaknesses of the authentication mechanisms, there appear to be several resources that are accessible without authentication. Specifically, these are located at /cgi/common.cgi, cgi/strminfo.cgi and a folder at /api that appears to hold a Java program. Overall, these files don't provide vast amounts of information to an attacker. The common.cgi file is perhaps the most interesting, as it contains the current internal IP address of the camera, as well as its mac address, current name, version information and gateway address (all important information for an attacker wishing to conduct a MITM-style attack).

Independently, these vulnerabilities wouldn't be considered particularly devastating. That said, combining several together does create quite a concerning attack. As an example, a malicious user may determine the current firmware version of a particular camera using unauthenticated access to the common.cgi file. From here, they may choose to initiate a brute force attack against a camera operating with non-current firmware (ie v1.07b5 and prior). Knowing the password is a maximum of eight characters, this would arguably be quite effective. Combining this process with a scanning tool, it would be possible to automate this entire process.

The wireless schemes employed by the device are also somewhat vulnerable. The camera itself contains wireless capability in two forms: infrastructure and ad hoc. Infrastructure mode uses an available router to connect to a network, allowing a user to access the camera through the router. Alternatively, the device can also be placed in ad hoc mode, in which it creates its own wireless network. A user can then connect directly to the device. Unfortunately, this ad hoc mode appears to only support WEP encryption (as opposed to WPA or WPA2). There have been several, well documented attacks against the security of WEP encryption within the literature (Stubblefield, et al., 2001; Cam-Winget, et al., 2003).

Post Authentication Attacks

Several different attacks and attack avenues are open to a user with authenticated access. Direct access to sensitive information is one such example. The web-based interface that a user would typically interact with appears to source information from 'hidden' cgi files. These files appear to be directly accessible by url, assuming the url is known by the interested party, and that party is authenticated. Reverse engineering the firmware, as well as investigating the strings stored within the main web server executable, reveal the location of these resources. They typically store information such as FTP details, the username and password of any attached email accounts, network details, a plaintext version of the current admin password and the wireless password to access the device (if applicable).

The server also appears to create a custom error message if a resource is not found (ie. It will state to the user that a particular resource does not exist). Although this does require authentication, it can be used to enumerate the resources on the server as a whole. For example, creating a script to iterate through possible resource names could be used to uncover further 'hidden' resources.

The server appears to employ client-side argument validation via JavaScript on several forms. By interacting with a request outside of the browser (eg. Through burpsuite) it is possible to bypass this validation. Whilst the server does seem to perform some additional validation itself, certain requests can still produce unwanted behaviour in the IP camera. For example, sending a request to change the name of the camera to no value (ie. Null or empty) results in the device becoming inaccessible. This will occur until a hard reset is performed, which will return the camera name to a non-null value.

Likewise, it is possible to add a user to the system with no name, through a similar process. This is actually somewhat more dangerous in terms of security. For starters, this username is hard to detect in the user list, given that it is simply a blank line. Additionally, it is not possible to delete this username from the list of users. This may not seem too unfortunate, although it does open the camera up to a persistent threat. If an attacker were to set the password of the blank user to null (or empty), certain parts of the camera will then be accessible without authentication (or more correctly, by authenticating with a username of null and password of null). Admittedly, if an attacker has the ability to add users to a system, they are already in a powerful position. The inability to remove the user, however, as well as the camouflage it employs, increases this threat somewhat.

The sanitisation of information from all sources appears to be quite thorough by the web server. There were no opportunities discovered that allowed cross site scripting (either reflected or stored). There was also no apparently ability to perform url path traversal, using encoded or non-encoded combinations. Although some functionality was distorted (eg. The camera name change), this effect was limited in scope and impact.

CONCLUSION

Overall, the IP camera in question contains moderate levels of security. Although it does exhibit some weaknesses, the lack of evidence for authorisation bypass or remote access is promising. It appears that the vulnerabilities discovered earlier in its lifecycle have been repaired, and have

ultimately acted to increase the strength of the device as a whole. Additionally, the proper sanitisation of variables by the embedded web server is considered effective.

That said, those weaknesses identified can still be used to impact a user. To this end, several recommendations are made to mitigate or prevent these vulnerabilities. Firstly, the firmware on the device should be upgraded to the latest version. Although up-to-date patching is important for all security products, this is particularly salient for the DCS-930L, given the susceptibility to brute force attacks that earlier firmware contains.

Additionally, the ad-hoc wireless mode should be avoided wherever and whenever possible. This is given the large amount of information in the literature pointing to weakness in the WEP encryption scheme. WPA2, whilst not perfect, is considered to be safer than WEP for wireless transmissions.

Perhaps the strongest threats against the camera come from an eavesdropped on the local network. There has been evidence to suggest a myriad of instances where highly sensitive information (such as passwords) is sent in clear text across the network. It appears that the cameras security will only be as strong as the network that it's placed upon.

There is some possibility for the DCS-930L to be used as an entry point onto the network. Assuming authorised access can be gained, the plain text information stored on the camera may give further access to the network. For example, the credentials for the FTP server are stored in the clear. Threats such as these should be considered when assessing the application of an IP camera. In practice, it would be wise to isolate this information to the camera alone, assuming that at one stage it will be accessed by malicious actors.

One area of the methodology that was perhaps at fault was the measurement of network traffic, especially in relation to the SSL communication between the D-Link website and the camera. Wireshark itself was monitoring from the PC that was interacting with the device. Whilst this would have been effective at capturing traffic between the two, it may have missed traffic going from the device, direct to any other sources (for example, directly to the D-Link website). In future, it would be advised to set the listening device on the router directly, to attempt to capture all traffic from the camera.

In summary, the DCS-930L is not without some vulnerabilities, but overall is suitable for the market it was designed for. Given the cost of the device, this is particularly true. If this camera were to be placed in a secure environment, however, the threats of local eavesdroppers and brute force would have to be strongly considered. It also offers further evidence and reasoning for why up-to-date patching of software and firmware is so vitally important in all security applications.

REFERENCES

Cam-Winget, N., et al. (2003) Security Flaws in 802.11 Datalink Protocols. Communications of the ACM 46, 35–39

D-Link (2013). "DCS-930L Firmware Release Notes." From ftp://ftp.dlink.ru/pub/Multimedia/DCS930L/Firmware/ DCS-930L-A1-release%20note-FW_1.08_build4.doc.

D-Link Australia (2013). "DCS-930L Wireless N Cloud Network Camera." Retrieved 1st of November, 2013, from http://www.dlink.com.au/products/?pid=889.

Doyle, J. (2012). "Password Disclosure in D-Link Surveillance Cameras (CVE-2012-4046) ". Retrieved 1st of November, 2013, from http://www.fishnetsecurity.com/6labs/blog/password-disclosure-d-link-surveillance-cameras-cve-2012-4046.

Group, N. W. (1999). HTTP Authentication: Basic and Digest Access Authentication. Retrieved from http://tools.ietf.org/html/rfc2617.

Heffner, C. (2013). Exploiting Network Surveillance Cameras Like a Hollywood Hacker. BlackHat, Las Vegas.

Heffner, C. (2013b). "binwalk: Firmware Analysis Tool ". Retrieved 1st of November, 2013, from http://code.google.com/p/binwalk/.

Paleari, R. (2013). "D-Link DCS Cameras Authentication Bypass / Command Execution." Retrieved 1st November, 2013, from http://packetstormsecurity.com/files/119902/D-Link-DCS-Cameras-Authentication-Bypass-Command-Execution.html.

PortSwigger (2013). "Burp Suite." http://portswigger.net/burp/.

Rocha, M., et al. (2013). "D-Link IP Cameras Multiple Vulnerabilities." Retrieved 1st of November, 2013, from http://www.coresecurity.com/advisories/d-link-ip-cameras-multiple-vulnerabilities.

Shodan (2013). "Shodan." http://www.shodanhq.com

Sood, A. K. and B. Gajbhiye (2011). Design Flaws in IP Surveillance Cameras: Exploiting Web Interfaces. Hackin9. http://www.cigital.com/papers/download/design_flaws_IP_surveillance_cameras_aityaks_bipin.pdf.

Stubblefield, A., et al. (2001). Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. http://taz.newffr.com/TAZ/Reseaux/Techniques_Attaques/wireless/wep_attack.pdf.

Wireshark Foundation (2013). "Wireshark: Whats on your network?". Retrieved 1st of November, 2013, from http://www.wireshark.org/.